(12)

NATURAL LANGUAGE GENERATION

Final Report

September 1, 1987 – May 31, 1991

Eduard H. Hovy

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695

Report of research
Sponsored by the Defense Advanced Research Projects Agency (DoD)
Technical Office: DARPA-ISTO
Project Title: Natural Language 2
ARPA Order No. 6096
Issued by Dr. J.A. Sears under Contract # MDA903-87-C-0641.

Contents

1 High-Level Project Objectives

Penman is a natural language sentence generation program being developed at USC/ISI (the Information Sciences Institute of the University of Southern California). It provides computational technology for generating English sentences and paragraphs, starting with input specifications of a non-linguistic kind.

The research objectives underlying Penman are threefold: to provide a useful and theoretically motivated computational resource for other research and development groups and the computational community at large, to provide a framework in which to conduct investigations into the nature of language, and to provide a text generation system that can be used routinely by system developers. Penman is being used by computer scientists (as the output medium of their programs, among others, projects in human-computer communication, expert system explanation, and interface design) and by linguists (as a reference and research tool).

This enterprise can be divided into two parts: the development of single-sentence generation technology and the development of multisentence planning technology. The Penman project has made a significant achievement with the first part and has recently started making great progress with the second.

During the course of the funded period it became clear that the Penman project could not survive unless it broadened its capabilities significantly. Providing the most expressive computational grammar of English for generation, in the form of an easily usable and widely distributed NLP language generation system, is not enough. We decided to focus some effort over the next two years on developing a prototype parser, and to link with other projects outside USC/ISI in complementary and collaborative enterprises such as Machine Translation, where Penman could provide useful service while building up strength in other areas of expertise.

2 Technical Research and Development

The work performed under this funding is described in 5 sections:

- Sentence generation
- Multisentence text planning
- Parsing
- Information retrieval
- Machine translation

2.1 Sentence Generation

During 1987, we completed a collaboration with BBN Inc. of Cambridge, MA, in the English-in English-out data base question answering system JANUS. In May 1987 the joint BBN-ISI data base question answering system JANUS was demonstrated at a DARPA meeting in Philadelphia. The JANUS parser was developed by BBN; the language generation was handled by Penman. The internal representation language and domain model (the model contained information about Navy ships, statuses, and actions) were shared by the two programs. A number of types of queries were parsed and a number of types of responses were generated, demonstrating the feasibility of English-in-English-out technology. At this point the collaboration had run its course, and both partners moved into different areas of research.

Additional regions of continuing interest and research were the extension of the expressive range and capabilities of the Nigel grammar and the streamlining and speeding up of the Penman program. A number of extensions to the Nigel grammar were completed during 1987. Some of these extensions were prompted by the new demands being made on Penman as a result of generating multisentential text: notably, the ability to generate interclausal linking expressions such as "in order to".

In work on lexical choice, several project members collaborated on developing an algorithm for more sophisticated lexical choice than had been implemented before. A description of this work has been accepted for presentation at the lexicon workshop to be held early 1988 in Boston.

The Nigel grammar was distributed to three sites — Columbia University, the Machine Translation Center at Carnegie-Mellon University, and (in paper form, due to licensing problems with IBM) to the IBM natural language research center in Los Angeles. We started actively searching for other suitable sites to which to send Nigel, both in the U.S. and abroad. In order to support the distribution of Nigel, we continued writing the Nigel documentation, an exhaustive description of the more than 500 grammatical systems in the grammar.

During 1988, Penman was distributed to the following sites: University of Saarbrücken, University of Delaware, University of Toronto, Glendon College (Toronto), University of Alabama (Huntsville), University of Illinois (Urbana). During this time, we completed the first release of the Penman documentation, which constitutes a set of four books: the Penman Primer, the Penman User Guide, the Penman Reference Guide, and the Nigel Manual. Though over 350 pages (of which about 250 are low-level grammar descriptions), this documentation is not yet complete. The Primer and User Guide were sent out to DARPA for review. The documentation has also been sent to the abovementioned distribution sites. In addition, as described in the next subsection, the input notation SPL was brought to implementation and actual use. It proved very flexible and easy to use. This work is described in the Penman documentation. Finally, the principal ancillary information source to Penman, the Upper Model, was subjected to detailed examination and reorganization, in order to reconcile a number of additions that had been made during the previous 18 months. In addition, a program was developed for helping new users of Penman link their domain definitions to Penman's Upper Model in a straightforward way. This work is described in the Penman documentation.

During 1989, the Penman project entered into two international collaborations, one with a research group in West Germany and one with the Department of Linguistics at the University of Sydney, Australia. Under the three-year cooperative agreement with the Integrated Publishing and Software Institute IPSI, a subgroup of the GMD (an umbrella organization for Computer Science and Mathematics research funded by the Federal German Government), IPSI will sponsor up to two person-years' work of Penman project researchers in return for the collaboration and use of Penman and associated natural language processing technology and expertise. The first three months of this collaboration began this quarter, when Dr. John Bateman flew to Germany in October to work there on building a German grammar for Penman and on overseeing the work of two graduate students on the formalization of various aspects of the grammar and on the possible addition of structural information to the grammar.

Also during 1989, we sent licensing agreements for the acquisition of Penman to the Computing Research Laboratory, Las Cruces, NM, and to UC Berkeley (Computer Science Department). We sent out a copy of Penman to Arthur Anderson Research Center, Chicago, and the new release of Penman to IPSI, the University of Sydney, and the University of Delaware.

In another new development in 1989, to enable the wider distribution of Penman (especially to linguists who cannot afford larger computers), Penman was ported to the Macintosh. Additional software was developed to make McPenman easy to use and extend, given that the Mac version will be the prime workhorse for the number of Systemic Linguists who have expressed interest in adding their own subgrammars to the central Penman grammar.

In 1990, having completed the porting Penman onto the Macintosh-II, we ported Penman onto Sun computer systems within two weeks (excepting, as always, the window interfaces; we lack the funding to hire the skills of an X-window specialist). The Sun version of Penman was used in the NLP course at USC as part of a large class assignment, and will in due course be distributed to various research sites who have asked for the Sun version of Penman.

During 1990, at their request, we sent licensing agreements for the acquisition of Penman to the following new sites: University of California at Irvine, University of Montreal, University of Sydney (Computer Science Department), CMU (Computer Science Department), Stanford University, and British Telecom Research Laboratories. We remain willing to send Penman to other suitable sites, both in the U.S. and abroad. We have also sent new releases of Penman to a number of sites, including the Computing Research Laboratory, Las Cruces, NM, and to UC Berkeley (Computer Science Department), University of California Irvine, CMU Center for Machine Translation, Stanford University, Philips Research Laboratories, and Rice University. At the end of 1990, Penman was distributed to over 45 sites in the U.S., Canada, Europe, and Australia. A first draft of the User's Manual for the Macintosh version of Penman was completed.

Also by the end of 1990, Penman ran on TI Explorer and Symbolics Lisp machines, on the Macintosh-II, and on Suns. The only exception is the window interface (which requires our hiring a specialist in X-windows for a month, something we could not under present circumstances afford). We continued to search for an opportunity to have someone else perform this task. In addition, given that Penman's window interface, and the auxiliary knowledge acquisition programs UPPERMOST and LAPITUP, require window interface management systems that differ across machines, we have embarked on a low-level ongoing effort to improve the interfaces when used in teletype mode, so as to provide useful functionality to users when porting Penman to new domains even when they do not use windows.

During early 1991, the German grammar being developed at Penman's companion project at the IPSI institute in Germany was continued. The person developing the grammar spent two months at ISI in order to build German grammar in precisely those areas required for the MT project domain texts.

2.1.1 Making Penman Easier to Use

During 1988, it became increasingly clear that the expressive power and range of Penman made the system difficult to use, since to harness that power the user had to provide much specialized information. In response, we spent much of the last quarter addressing this issue. Our solution was to divide the information Penman requires to produce sentences among various sources of input, and to enable the user to default away or to define in very simple ways most of this input. In particular, we developed an input language called SPL that replaced the former input language and that supported the following features:

- recursive frame-like input specifications
- default facility
- macro definitions

The development and implementation of the input notation SPL to replace the previous input notation formed the major technical achievement of this quarter.

SPL is a notation in which users of Penman, including text planning programs, can specify plans for sentences at various levels of abstraction and various amounts of detail. These plans serve as input representations for Nigel, Penman's sentence generation program, and they can be regarded as specifications of constraints that must be satisfied by the generated sentence. SPL representations are lists of terms describing the types of entities and the particular features of those entities to be expressed in English. The features of SPL terms may be either semantic relations to be expressed from the application's knowledge base, or responses to Penman's inquiries which determine linguistic attributes of sentences.

Because SPL representations may contain both linguistic and non-linguistic features, they are able to specify constraints on how something is expressed, when necessary, in addition to specifying the propositional content that is to be expressed. The linguistic features of SPL may be used to produce sentences that exhibit a style appropriate for a particular application domain, as well as providing control over common linguistic variations, such as thematization, passivization, and type of modification. In order to provide control at varying levels of detail, SPL accommodates partial specifications. The information in a specification may be augmented by merging information from coreferential terms and by applying sets of default feature values that may be modified dynamically by the application.

An important feature of SPL is thus the ability to declare and set collections of default values at any level of specificity, from user-defined domain-specific aspects to grammatically-oriented linguistic features. Our implementation supports the hierarchical stacking of default settings, which enables a set of Penman-supplied defaults to underlie all user-supplied defaults and act as a safety net when the user oversteps the bounds of his or her expertise.

A second important feature of SPL is the ability to define new features to augment the language. Many features required to control Penman's grammar Nigel are germane to the production of English but are not of particular interest in the user's domain of application. However, collections of these features frequently do play a role in the user's domain, and in order to save the user the trouble of having to investigate and control the individual features wherever required, the SPL macro facility enables the user to bundle such feature collections together and give them a name that is germane in the domain. For example, to produce sentences in English, one requires three times: the time of speaking, the time of the event being discussed, and the time of the vantage point being taken toward the event. Such complexity is seldom required by real computer applications, most of whom are content with a simple three-way present/past/future distinction. Penman's macro facility enables the user to define a feature called, say, TENSE, that takes one of the three values, and that expands into the appropriate features and values for the Nigel grammar.

The first implementation of the language SPL has been completed and tested to some degree. It has proved very flexible and useful, to the point where ISI-internal users of Penman, who previously relied on code written by members of the Penman project to rewrite their inputs into the language required by Penman, are now producing SPL plans themselves.

A paper by Kasper, R.T. and Whitney, R.A. describes the structure and implementation of SPL: *SPL: A sentence plan language for text generation.*

2.1.2 International Penman Collaboration

During late 1990, we hosted at USC/ISI the principal members of the two other sites of the international Penman team: Prof. Christian Matthiessen from the University of Sydney in Australia, a graduate student from there, and Dr. John Bateman, currently on leave from ISI and heading the NL generation team at the IPSI research institute in Darmstadt, Germany. Also present was Dr. Bob Kasper from the Ohio State University, who used to be on Penman and is still directing the parser development. We held a day-long "Penman summit" to coordinate future research across the various sites, in particular to plan out mutual development of resources for MT (including German, Japanese, and Chinese grammars for Penman), and also the sharing of people (especially students) across various sites.

2.2 Multisentence Text Planning

The analytical work describing the structure of on multisentence text conducted since 1983 by Dr. William Mann (in collaboration with Prof. Sandra Thompson from UC Santa Barbara) in developing the Rhetorical Structure Theory (RST) was taken a step further in 1987 when Dr. Eduard Hovy joined the project and built the first text structure planner using RST relations as text plans.

The multisentential text planning technology was linked to Penman and to two disparate application domains, resulting in Penman's generating paragraphs of text for a multimodal Navy data base information display system and a self-explaining expert system.

In their work on Rhetorical Structure Theory, Dr. Mann and Prof. Thompson had analyzed hundreds of paragraphs and proposed that a set of 20 relations suffice to represent the relations that hold within texts that normally occur in English. These relations are interpreted recursively; the assumption being that a paragraph is only coherent if all its parts can eventually be made to fit under one overarching relation.

Under a slightly more operational and goal-oriented interpretation, these relations can be seen as plans that govern the assembly of clauses into coherent paragraphs. Dr. Eduard Hovy developed a top-down hierarchical planner similar to the well-known planner NOAH which employed these relation/plans to structure paragraphs from given collections of input.

This method of planning paragraphs afforded much more flexibility of assembly than previous methods allowed. In this sense, the development of multisentential planning technology paralleled the development of single-sentence generation technology. The earliest techniques for producing single sentences by computer relied on the canned text and template methods, in which either a predefined string stating the information was selected and output, or slots of a predefined template were filled in with appropriate aspects. Though useful in very limited domains, the lack of an intelligently controlled construction process imposed severe limitations on the flexibility and extensibility of such techniques. In the last decade, work on sentence generation has produced progressively more refined generators, to the point where the most powerful generators today dynamically assemble large numbers of very detailed grammatical features which together specify a sentence.

The work on the production of multisentential paragraphs has had a similar history, with a roughly 10-year lag time. The first systems to produce paragraphs of text used so-called schemas that described the content and order of the clauses of the paragraph. Each schema was a static representation of a particular discourse strategy that people typically employ in conversation; each schema thus produced a different type of paragraph. Though early schemas afforded some variation, and later schemas were built to accommodate additional types of variation, these structures in general suffer from the same lack of flexibility that hampered sentence templates.

During the last year, the Penman project has been formalizing RST relations and using them generatively to plan paragraphs. Relations are seen as plans — the operators that guide the search through the permutation space of the input units. Constraints on the parts of the relation/plans become requirements that must be met by any piece of input before it can be used in the relation (i.e., before it can be coherently juxtaposed with the preceding text). The effects of relation/plans are descriptions of the intended effect of the relation (i.e., the communicative goal that the relation achieves, if properly executed). Since the goals in generation are communicative, the intended effect must be seen as the inferences that the speaker is licensed to make about the hearer's knowledge after the successful completion of the relation/plan. The constraints and effects of plans are represented in terms of the formal theory of rational interaction currently being developed by, among others, Cohen, Levesque, and Perrault.

The text structure planner operates antecedent to Penman. It plans coherent paragraphs to achieve communicative goals posted by the user's system to affect the hearer's knowledge in some way. It accepts one or more inputs from the domain of discourse, rewrites the inputs into a common form (called here input units) which consist of collections of input characteristics, and by planning assembles the input units into a tree that expresses the paragraph structure. Finally, the planner traverses the tree, dispatching the leaves (the input unit clauses) to be generated by Penman. During traversal of the tree, additional planning tasks, such as sentence size delimitation and focus control, are performed.

The planner embodies a limited top-down hierarchical expansion planning framework. Each relation/plan has two parts, a *nucleus* and a *satellite*, and relates some unit(s) of the input or another relation (cast as nucleus) to other unit(s) of the input or another relation (cast as satellite) recursively. In order to admit only properly formed relations, nuclei and satellites contain requirements that must be matched by characteristics of the input. (Thus, for example, the PURPOSE relation/plan cannot be used to relate some input state or condition to some input action unless it can be proved (to the planner's satisfaction, using the PURPOSE requirements) that the state was in fact the purpose of the action.)

The multisentential planning technology was initially tested on two distinct application domains: Integrated Interfaces (II) and the Program Enhancement Advisor (PEA).

2.2.1 Text Structure Planning for II

The Integrated Interfaces project at ISI is developing a prototype interface management system that handles mixed mode input (menus, forms, pointing), and incorporates a combination of output modes (NL text, maps, menus and forms). Integrated Interfaces uses Artificial Intelligence knowledge base and rule technology to link together knowledge of an application domain with facilities of the user interface. A frame-based knowledge representation system (LOOM or NIKL) is used to model the entities of the application domain and the facilities of the user interface. These domain and interface models are related by antecedent-consequent rules to determine appropriate methods of displaying information.

The II project has implemented a demonstration interface to an existing Naval database reporting application. This interface system creates displays similar to those being prepared manually for the Navy on a daily basis. In these displays, Penman generated natural language texts, which were placed at appropriate locations on a map to describe the activities of important objects, such as ships.

Within Integrated Interfaces, our aim was to provide a component that would present, in natural language, the information which was suited to language and not suited to two-dimensional display modes such as maps or tables. Our task was to accept from the II display manager the information to be generated, to structure this information into coherent paragraphs, and to generate the English sentences comprising the paragraph. The information is provided in the same representation scheme used by the rest of the II system; no special language-based alterations are made. Thus the Penman text structure planner and sentence generator act as a subsystem of II. Our task consisted of three principal parts:

1. the prestructuring of the input into individual clause-sized units;
2. the construction of a coherent paragraph, including appropriate interclausal relation words; and
3. the generation of individual clauses in English.

Task 1 required domain-specific information. We built a module that used Navy rules to group the information provided by II into clause-sized units and to extract from the information such events as arrivals, departures, and rendezvous events (this information is not explicitly represented in II. n order to be generated, however, it must be given explicit status. For example, arrival events were created using the rule [if a mobile employment is followed by a stationary employment, then an ARRIVE event occurs between them]).

For task 2, the text structure planner was used to build a coherent paragraph from the clause-sized units. As described, RST relation/plans provided the constraints necessary to impose coherence and also provided typical interclausal conjunctive words and phrases where useful.

For task 3, a tree traversal algorithm gathered the clause-sized chunks of input contained in each leaf of the tree and activated Penman with these chunks to produce a list of sentences. The paragraph of sentences was then returned to the Integrated Interfaces display manager to be formatted and presented on the screen inside a text block. The following paragraphs are a few of those generated for this domain:

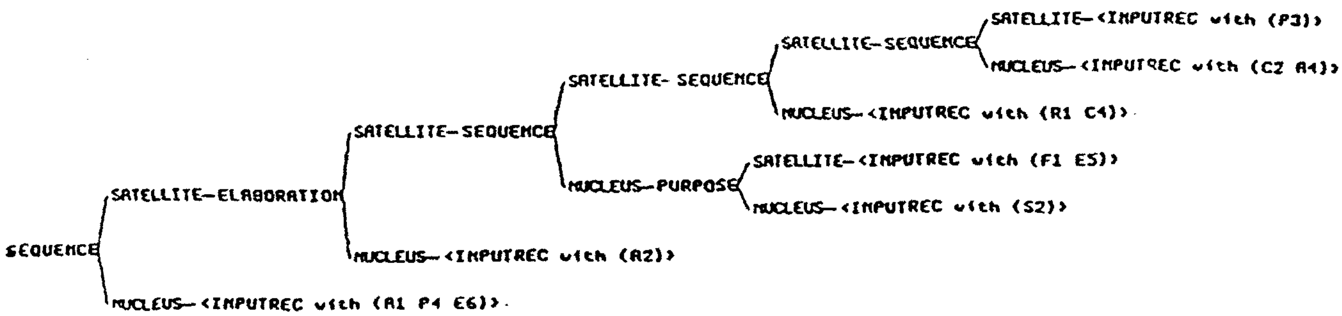
```
Knox, which is C4, is at 79N 18E heading SSW. It is en route to
Sasebo, arriving 4/24, in order to load for four days.

Knox is en route to Sasebo. It will arrive 10/24. It will load
until 10/28.

Knox and Fanning are en route to Sasebo, arriving 4/24. While
it is in Sasebo, Knox, which is C4, will load until 4/26.
Fanning will depart on 4/25 in order to rendezvous with CTG 73.1
on 4/28.

Fanning, Passumpsic and Whipple are en route to rendezvous with
CTG 070.10, arriving tomorrow. Fanning and Whipple will be on
operations until 10/26. Passumpsic will be performing services
until 10/28.

New Jersey, Copeland and Merrill are on operations until 4/20.
Thach is on operations until 4/21.

MEKAR-87 takes place with Knox, Fanning, and Whipple in South
China Sea from 10/20 to 11/13. Knox and Fanning join 10/20.
Whipple, which is C4, joins 10/29. Knox departs on 10/31.
Fanning and Whipple depart on 11/13.
```

The result was very successful. The Integrated Interfaces system was demonstrated to a number of Navy officials, among others the personnel responsible for creating the Navy's daily briefings, which consist of maps showing ship positions and descriptions of their employments. They currently perform this task by hand. Their reaction was extremely positive, to the point where funding is currently being negotiated to produce a full-scale system for use by the Navy.

From our perspective, we found the Integrated Interfaces experiment very useful in identifying the difficult problems inherent in paragraph planning, and in finding the points where further theoretical work is indicated. For example, we believe that the planning of page-length reports of the activities of ships and of the visitors to ports is now a feasible undertaking. This would be the first time ever that a computer has planned and generated more than two or three relatively short paragraphs of connected text.

2.2.2 Text Structure Planning for PEA

The second testbed for the multisentential planning technology was an expert system, part of the Explainable Expert Systems project. The Program Enhancement Advisor expert system (PEA) suggests improvements to Lisp programs and interactively explains its recommendations and its reasoning to the user. We are collaborating with PEA in order to produce text that is satisfactory for its needs. We have not yet completed the modeling of the domain knowledge in terms Penman understands; however, to date we have produced a handful of paragraphs, of which the following text and associated paragraph structure tree is an example:




```
The system asks the user to tell it the characteristic of the program
to be enhanced. Then the system applies transformations to the program.
In particular, the system scans the program in order to find
opportunities to apply transformations to the program. Then the system
resolves conflicts. It confirms the enhancement with the user. Finally,
it performs the enhancement.
```

2.2.3 Continued Text Planner Development

During 1988, multisentence text planning support was provided to the two expert systems then being built by the EES project, namely the Program Enhancement Advisor (PEA) and the Digital Circuit Diagnosis (DCD) systems. Both systems benefited from the use of the new SPL input notation. In both cases, the domain specialists are now creating their own input notation without having to use code written by members of the Penman team. As a result, they were able to generate more text at a quicker rate. Reports of this appear in two papers submitted to the 1989 ACL conference (papers by Bateman and Paris and by Moore and Swartout). Similarly, text was generated for the Integrated Interfaces project during 1988. During this year, Dr. Hovy collaborated with Prof. McCoy from the University of Delaware in order to incorporate her theory of focus shift in the text planner.

A final burst of support was provided for one of the subprojects of the EES project during 1989. This subsystem, the Program Enhancement Advisor, ended in December, when its author, Dr. Johanna Moore, graduated from UCLA and took a job at the LRDC and University of Pittsburgh. Penman provided all the linguistic functionality for PEA.

During 1990, we implemented several new text plans to handle texts from the ARIES Air Traffic Control domain program development system (being developed at USC/ISI under RADC funding by Dr. Lewis Johnson and his group); we demonstrated the planner to Mr. Doug White from the RADC, who monitors our text planning grant. extending this work, we developed theory and then partially implemented in the Penman text structure planner the ability to include text formatting information. When completed, this will enable the planner to produce, from computer-internal representations, paragraphs of text *with appropriate formatting instructions.* A prototype version with one such formatting instruction (enumeration) was implemented and shown in an example.

The ISI-IPSI text planning group designed and partially implemented a new text planner during the middle 7 months of 1990. Members of this group included Drs. Hovy and Paris from ISI, Mr. Mittal from ISI, and several visitors to the project: Dr. Julia Lavid, a text linguist from the University Complutense of Madrid, Ms. Elisabeth Maier, a member of the IPSI Penman group in germany, and Mr. Giuseppe Carenini, from the IRST research institute in Trent, Italy.

2.3 Parsing

During 1988 and 1989, under AFOSR funding, Dr. Robert Kasper has adapted a parser, built as part of his dissertation work, to use Penman's grammar Nigel (see attached paper). The first version of this parser was built by extending PATR-II, a general unification-based system. It demonstrated that a general parsing capability could be developed for systemic grammars that are expressed in a declarative notation.

Having performed on this work, given the underlying formal equivalence between unification and the subsumptive relation, the idea was conceived of implementing the parser itself *in Loom*, using Loom's subsumptive classifier as the central inference operation instead of unification.

This idea is appealing on several grounds. The most exciting aspect is the fact that for the first time, semantic and syntactic parsing will be able to be performed simultaneously, in one same representation system, using the same underlying operation. In traditional parsers, semantic and syntactic processing are performed separately, requiring a great deal of bookkeeping to ensure that the underlying interdependencies are recognized and fully utilized in the mutual disambiguation process.

However, since Penman's semantic representation — the Upper Model and Domain Model — are represented in Loom, and since its grammar, which is a systemic network, can easily be represented in Loom, both semantic and syntactic information can be put into the same representation system. This is one step toward a fully integrated semantics-syntax parsing paradigm often seen as the ideal.

Another step relates to the actual parsing operation. Since Loom is a member of the KL-ONE family of knowledge representation languages, it provides a classifier which is able to locate the most specific concepts that subsume an arbitrary set of features. This classifier will be used by the parser as the central inference operation; each word will be read by the parser and classified with respect to both syntactic network (the grammar) and semantic network (the Upper and Domain Models), based on its lexical features and position in the sentence. Similarly, intermediate structures will be incrementally built up and classified, until finally the overall structure(s) is fully classified, constituting the parse.

In summary, developing integrated syntactic and semantic parsing in Loom requires a number of steps:

1. Loom has to be given the ability to perform inference over disjunctions. That is, since midway during a parse, the system typically has to maintain a number of possible alternatives, Loom has to know how to reason about an instance being subordinate either to one concept or to another but not to both, and how to go about collapsing disjuncts when the infeasibility of all but one of its disjuncts become known.

2. Penman's grammar has to be represented in Loom. Code must be written to transform its current form, namely a fairly straightforward Systemic Network, into a set of Loom assertions that preserve all the properties of the grammar.

3. The underlying interdependencies between the grammar and the semantic models must be captured in Loom terms. This requires the formalization of some of Penman's inquiries.

4. The parsing mechanism must be rewritten to interact with the Loom classifier, and the results of the parse must be assembled and presented in a standard tree-like form.

2.3.1 Parsing in Nigel

This section describes a new approach to parsing that utilizes recent advances in unification-based parsing and in classification-based knowledge representation. This work is part of an effort to provide the Penman system with full natural language input and output capabilities. An experimental prototype of this parser using unification and a feature structure representation of part of Penman's grammar has been completed successfully. Most of the work in constructing a parser using the classification-based architecture of Loom and to reproduce the functionality of the unification-based system, now operating on the whole of the grammar, has been completed.

This experiment appears to provide a way of substantially reducing several of the most general sources of inefficiency that are observed in current unification-based parsers. However, this conjecture needs to be examined by performing experiments with several real grammars and applications. In addition to providing an efficient engine for processing the constraints of linguistic feature descriptions, we also expect this type of information organization to provide a strong basis for integrating semantic knowledge and knowledge specific to particular applications into the parsing process.

As unification-based grammatical frameworks are extended to handle richer descriptions of linguistic information, they begin to share many of the properties that have been developed in KL-ONE-like knowledge representation systems. This commonality suggests that some of the classification-based representation techniques can be applied to unification-based linguistic descriptions. This merging supports the integration of semantic and syntactic information into the same system, simultaneously subject to the same types of processes, in an efficient manner. The result is expected to be more efficient parsing due to the increased organization of knowledge.

The use of a KL-ONE style representation for parsing and semantic interpretation was first explored in the PSI-KLONE system [Bobrow & Webber 80], in which parsing is characterized as an inference process called *incremental description refinement.* The key idea underlying this process is that a description of an object can become increasingly more specific as additional features are learned from multiple knowledge sources, which is essentially the same idea that underlies most unification-based approaches. Bobrow and Webber identified four crucial capabilities that a representational system should have in order to support the process of incremental description refinement. These capabilities, not all available to Bobrow and Webber in 1980, have recently been developed in the Loom knowledge representation system [MacGregor 88] and hence enable the practical development of the new parsing method. They are:

1. Determination of the properties of a structured object that provide sufficient information to guarantee the applicability of a description to (some portion of) that object — i.e., criteriality conditions. Loom provides a separation of definition (necessary and sufficient conditions) and constraints (implied features).

2. Determination of the mappings that are possible between classes of relations — e.g., how functional relationships between syntactic constituents map onto semantic relationships. This is not part of Loom, but can be captured in the interrelationships between a syntax-oriented grammar and a semantics-oriented concept taxonomy.

3. Determination of the pairs of descriptions that are mutually incompatible — i.e., which cannot both apply to a single individual. Loom provides more complete inference of disjointness than previous systems in the KL-ONE family.

4. Determination which sub-categorizations of descriptions are exhaustive — i.e., at least one of the subcategories applies to anything to which the more general description applies. Loom provides inference with respect to coverings, implemented by disjunctive descriptions.

Constraints in Unification-based Grammars A variety of current approaches to parsing in computational linguistics emphasize declarative representations of grammar with logical constraints stated in terms of feature and category structures. These approaches have collectively become known as the unification-based grammars, because unification is commonly used as the primary operation for building and combining feature structures. Some of the simplest of these grammatical frameworks, as exemplified by the PATR-II system [Shieber 84], state constraints on features entirely in terms of sets of unifications that must be simultaneously satisfied whenever a grammatical rule is used. In such systems all constraints on a rule or lexical item are interpreted conjunctively. Many of the more recent frameworks also use other general logical connectives, such as disjunction, negation and implication, in their representation of constraints. The utility of such logical constraints is abundantly illustrated by linguistic models, including Systemic Grammar (SG) [Halliday 76] and Head Driven Phrase Structure Grammar (HPSG) [Pollard & Sag 87], and by computational tools such as Functional Unification Grammar (FUG) [Kay 85]. For example, SG and FUG even use disjunctive alternations of features, instead of structural rules, as the primary units of grammatical organization. While the intuitive interpretation of these logical constraints is rather straightforward, and they are quite natural for linguists to formulate, large-scale implementations of them have typically involved finding a balance between expressive power and computational efficiency, not an easy task.

Some difficulties can be expected in developing a system for computing with disjunctive and negative feature constraints, because it has been established that common operations on such descriptions, such as unification and subsumption, are NP-complete and require exponential time in the worst case [Rounds & Kasper 86]. The most common and obvious way to deal with disjunctive constraints is to expand the grammatical description to disjunctive normal form (DNF) during a pre-processing step, thereby eliminating disjunction from the rules that are actually used by the parser. Though this method works reasonably well for small grammars, it turns out to be unsatisfactory for larger grammars. Thus, several unification algorithms for disjunctive feature descriptions have been developed in recent years: [Karttunen 84, Kasper 87, Eisele & Doerre 88]. The Kasper algorithm was first implemented as an extension to the unification algorithm of the PATR-II parser, and it has been further developed to handle conditional descriptions and a limited type of negation [Kasper 88a]. These extensions to PATR-II have been used to construct an experimental parser for systemic grammars [Kasper 88c], which has been tested with Penman's grammar of English (which was developed primarily for language generation system [Penman 88]).

Although these methods for processing complex feature constraints are generally much more efficient than expansion to DNF, they still have several significant sources of inefficiency:

1. a large amount of structure must be copied in order to guarantee correct unification;
2. consistency checks are required between components of a description that do not share any features in common, because unification cannot determine whether any dependencies exist between two structures without actually unifying them;
3. repeated computations are often required over sub-expressions of descriptions, because the results of prior unifications (and compatibility tests) are not saved.

These sources of inefficiency are not unique to one method of parsing with disjunctive descriptions; similar shortcomings are commonly reported for most unification-based systems. The unification literature contains several techniques for reducing the amount of copying by structure sharing, but these techniques appear to solve only part of the problem. In response, we have adopted methods that are used in classification-based systems, which provide a more general approach to improving the efficiency of the parser. These methods are described below.

Classification-based Knowledge Representation Instead of using unification, we have found that it is possible to use classification, a formally similar operation, to avoid many of these inefficiencies. Two kinds of improvements are possible: first, since the components of the grammar are known before parsing commences, various relationships, such as subsumption and compatibility, can be used to construct a lattice of grammatical objects, eliminating the need to derive them repeatedly at parse time. Second, by retaining the results of matching and classification during the parse, multiple matching can be avoided.

Loom [MacGregor 88], which has been developed at USC/ISI, is a member of the KL-ONE family of knowledge representation systems, which are based on an explicit logical formalization of many of the constructs that have been explored in semantic networks and frame-based representation systems. They organize information about objects and the relations between them into hierarchies according to specificity, with more specific objects placed below more general ones. For example, a hierarchy of English word classes would probably contain Verbs, Transitive-Verbs as a subclass of Verbs, and the word "like" as an instance of Transitive-Verbs. Each hierarchy is a subsumption-ordered lattice based upon logical properties that can be deduced from the definitions of objects and the facts known about them. In these systems, classification is the operation that places a new class or object into the lattice according to the subsumption order. A primary benefit of classification is that it organizes large collections of knowledge in such a way that properties shared by many objects need only be represented once, yet they can still be efficiently accessed by inheritance.

The classification-based architecture used by Loom solves a whole class of related efficiency problems by explicitly constructing and maintaining a subsumption-ordered lattice with inheritance. In particular, these savings are:

Structure Sharing: Classification-based systems do not require copying of the entire structure under consideration, because the description of a constituent can contain pointers to the classes of objects that it instantiates. This representation not only saves space, but it also allows the parser to make use of information that has already been precomputed (during the classification process) for classes of objects in the grammar and lexicon. Hence the organization of descriptions into a lattice automatically provides a great amount of structure sharing.

Indexing Dependencies: The process of classification also keeps track of dependencies between different objects, eliminating the need for checking consistency between components of a description that have no features in common. In effect, an index is incrementally constructed from features to descriptions that contain them. This contrasts with most unification-based systems, in which feature structures are represented by directed graphs (or by first order terms, as in Prolog).

Avoiding Redundant Computations: With un-typed feature structures, each unification is performed on a pair of structures without reference to any stored knowledge, i.e., there is no way for simple unification to use the results of previous unification and subsumption computations, even if many objects with identical features have already been unified. By explicitly representing the types of objects in a lattice, information can be stored for classes of objects, making it possible to avoid repeated computations for multiple objects having the same type (or any more specific type). Thus the first time a component of a description is classified, it is placed into the lattice containing all other descriptions in the knowledge base. Since the lattice explicitly represents the types of objects, it makes full-depth consistency checks unnecessary between objects that are known to be in a subsumption relationship, and subsumption (success) and consistency (failure) tests only need be computed once for all objects that belong to the same types.

Using Classification as a Grammar Compiler: Finally, classification can be seen as providing a capability similar to that provided by compilers in programming systems. Although a simpler unification-based system may provide acceptable results with somewhat less overhead than a classification-based approach on a limited scale, a classification-based system is almost certainly to be preferable for applications that are necessarily knowledge-intensive.

An Experiment in Classification-based Parsing KL-ONE and similar frameworks have been used for semantic interpretation in some natural language processing systems [Sondheimer et al. 84], but usually in a way that is quite separate from the grammatical parsing process (an exception is the aforementioned PSI-KLONE system). Generally speaking, linguistic categories correspond to concepts, and their features (or attributes) correspond to binary relations in the knowledge representation system.

Many formal properties are shared by the feature descriptions used in unification-based grammars and the terminological definitions used in KL-ONE. However, despite the underlying similarities, there are significant differences in the expressive capabilities that are usually provided. In particular, the knowledge representation systems typically have general constraints on relations with multiple values, whereas most unification-based systems do not provide a direct representation for features with set values. On the other hand, complex logical constraints involving disjunction and negation have been more extensively developed in unification-based systems than in classification-based systems. The Loom system appears to be the first in the KL-ONE family to have included general disjunction and negation in its concept definition language. The implementation of classification for disjunctive concepts has been based on several refinements of a strategy that was originally developed for unification with disjunctive feature descriptions [Kasper 87]. With these extensions, the Loom system is able to handle a much fuller range of constraints that have been used in actual linguistic descriptions of feature structures.

In order to explore a strategy for parsing based on classification, we have to represent Penman's grammar in Loom and replace the existing unification component of our parser (see [Kasper 88c]) with activations of Loom's classifier. Motivating this action are two primary goals: to investigate the extent to which classification can be used to organize the knowledge contained in linguistic descriptions so that it can be more efficiently accessed during the parsing process, and to develop a suitable architecture for integrating semantic information into the parsing process, in a way that knowledge specific to application domains does not have to be re-organized for parsing.

It is straightforward to convert the feature constraints of the grammar into a set of definitions that can be processed by Loom, because of the underlying correspondences between Loom's concept definitions and linguistic feature descriptions already described. It is also straightforward to perform an operation that is equivalent to the unification of feature structures within Loom. This is accomplished by forming an object having a type that is defined as the conjunction of the types corresponding to the feature structures.

Instead of unifying a partial description of a constituent with a grammatical description, the description of the constituent is classified with respect to an object-oriented representation of the grammar, in which each object stores information and constraints associated with a particular type of grammatical constituent. The classifier determines which grammatical classes the constituent instantiates, and the constraints associated with these classes can be used to give a more complete (grammatical, semantic, pragmatic) description of the constituent.

A Simple Example In an example, consider how classification with respect to a simple grammar may be used in parsing the sentence: *David likes computers.* Assume that a lexical/morphological analyzer gives the following type membership information for each word:

```
David: Noun.
computers: Noun.
likes: Verb Transitive Present.
```

Also assume that a rather simple context-free grammar can be used to recognize possible constituents, and that it can be annotated to assign grammatical functions[1]. In the example sentence, this grammar proposes a constituent c with the type Clause and the following grammatical functions:

```
subject : david
process : likes
dobject : computers
```

This initial description of the constituent, c, is then given to the classifier, which deduces the most specific types that it belongs to. The classifier begins by considering types that are directly below the initial type, namely, `Clause`. Assume the grammar specializes clauses into one of two types, either `Intrans-Clause` and `Trans-Clause`, where the former is defined to be a type of `Clause` with the role `process` whose filler must be of type `Intransitive`. This definition is not satisfied by c, because its process, `likes`, is not of type `Intransitive` in the action definition hierarchy. Next, the classifier considers `Trans-Clause`, which is defined to be a `Clause` with a `process` of type `Transitive`. This definition is satisfied by c. In addition, `Trans-Clause` is defined to have a constraint: it implies the type `Active OR Passive`, which means that any object which is a member of `Trans-Clause` must also be a member of `Active OR Passive` (that is, `Active` and `Passive` form a disjoint covering of `Trans-Clause`). Therefore, `Active OR Passive` is added to the list of types to which c belongs.

[1] Using the classification-based approach outlined here, it is theoretically possible to perform the parsing completely using only classification. However, such a parser would have to examine all substrings of the input in order to find all possible constituents, unless sufficient constraints on constituent ordering can be applied early enough in the parsing process. By performing a shallow structural parse before starting the deep classification-based parse, one gains a large improvement in efficiency, because even a skeletal context-free grammar can provide the basic segmentation of the input sentence into its major constituents. Thus, a simple context-free parsing component was used for this purpose with success in the prototype system.

Because `Active OR Passive` is a disjunction, it is possible to infer membership in one of the disjuncts by proving incompatibility with all other disjuncts. `c` is compatible with all of the constraints of `Active`, but it is not compatible with the constraints of `Passive`: it has a process of type `Present`, `Passive` requires a process of type `PastPart`, and the types `Present` and `PastPart` specialize the disjoint types, `Finite` and `Nonfinite`. By eliminating the `Passive` disjunct from consideration, membership in the `Active` disjunct can be inferred. `Active` is the most specific type that can be inferred for `c`, because it specializes all other types that `c` belongs to (and there are no more specific types defined in this simple example).

As a consequence of acquiring membership in the type `Active`, `c` inherits all constraints that are associated with `Active`. These constraints require that the `actor` and `subject` roles are identical (i.e., that the values of these two roles should be unified), and that the `goal` and `dobject` roles are identical. Satisfying these constraints yields the following information about the roles of `c`:

```
actor : david
goal : computers
```

Thus, given the initial assumption that `c` is a clause with particular constituents filling the grammatical functions `process`, `subject` and `dobject`, classification deduces a more specific type (that `c` is an active clause) and also values for previously unspecified roles (`actor` and `goal`).

The classifier uses the lattice representation of defined types to guide its search for types that are satisfied by a given object. It does not need to consider any types that fall below a type that the object is known not to specialize, such as all types below `Intrans-Clause` and `Passive` for the object `c`.

The power of using this kind of classification scheme may be further exploited by associating semantic and pragmatic constraints with each grammatical type, in addition to the grammatical constraints which have been illustrated.

Integrating Semantic Information into the Parsing Process One of the greatest advantages of this method of parsing is the possibility of performing integrated semantic and syntactic processing. KL-ONE systems such as Loom were traditionally developed to represent semantic information, and with the inclusion of syntactic information as required for the work described here, both types of knowledge reside in the same system and are accessible to a single classification process.

Traditional parsers either have a pipeline architecture, in which syntactic parsing precedes semantic parsing, which in turn usually precedes pragmatic parsing and anaphor treatment, or an interleaved architecture, in which the various aspects are interleaved. In both cases, the separation of processing according to underlying knowledge source complicates the process. In the pipeline model, ambiguities have to be remembered until later stages can disambiguate them, and in the interleaved model, complex bookkeeping is required in order to ensure consistency of the processes.

The parsing method proposed here is a radically different approach: the fully integrated use of syntax, semantics, and whatever other relevant knowledge can be represented in Loom. The classifier simply accesses all the relevant information, regardless of its conceptual type, and uses their interdependencies to resolve ambiguities in the natural course of its processing. By being able opportunistically to access both semantic and syntactic knowledge at any point during the process, the parser can resolve ambiguities sooner than in the traditional pipeline model, in which syntactic parsing is completed before semantic parsing commences. Many of the structural ambiguities that arise during parsing are only resolvable by semantic knowledge, and pipeline parsers have to maintain all the syntactic possibilities until the semantic parsing phase. Non-pipeline parsers have to perform a complex interweaving of semantic and syntactic processing, requiring increased bookkeeping and more complex system architecture. In the method outlined in this paper, the parser's single call to the classifier will result in the most appropriate information — both semantic and syntactic — being found and reconciled, if possible by *the normal action of the classifier.*

This integration exhibits a highly desirable simplification of process, reduction of processing overhead, and facilitation of representation of dependencies between syntax and semantics.

Another benefit is the increased portability provided by a knowledge representation paradigm used in the Penman system. In order to achieve greater portability, Penman contains a general taxonomic ontology of concepts called the Upper Model [Bateman et al. 90], under which the concepts from various application domains are subordinated. By inheriting information from the Upper Model, domain concepts can be handled appropriately by the Penman language generator without the generator ever having to be explicitly informed of their individual nature. Similarly, the parser can exploit inherited Upper Model information when trying to place words appropriately into structures. More information can be found in [Kasper 89a, Kasper & Hovy 90].

2.3.2 Continued Progress on Parsing

During 1990, Dr. Kasper continued extending Penman's parsing capabilities. Code was written to automatically reformulate Penman's grammar into a Loom representation, and work on linking the grammar to Penman's Upper Model was nearing completion. Apart from converting the basic parser to call Loom's classifier instead of PATR-II's unifier, these are the two steps remaining before the new Penman parser can commence testing.

Parser work during this time involved extending the capabilities of the Loom Knowledge Representation system to include the ability to handle multiple worlds, which is the mechanism in our design for keeping track of all the ambiguities that arise during parsing. Due to some shortcomings in the Loom code, this work took longer than we anticipated, but was completed by the end of January 1991.

During early 1991, parsing work continued under the supervision of Dr. Bob Kasper (Ohio State; consulting at ISI) to extend the capabilities of the parser and Loom. In the staged approach being followed, all the prerequisites for noun phrase parsing were completed.

2.4 Information Retrieval

This section describes a proposal written during 1990 to perform semantic-based information retrieval that was submitted to Darpa unsuccessfully. Because we believe the idea was essentially valid, and because we intend to pursue it in the future, we include it here. Co-authoring the proposal with Dr. Hovy was Dr. Patrick Jost from TRW Inc.

The proposal describes a novel combination of technology from three sites to perform **high-accuracy multilingual document retrieval and classification** from large collections of documents.

The proposed system exploits the strengths of two separately developed technologies by putting them together in such a way as to overcome their individual shortcomings. The result is an exciting and powerful document retrieval system that combines the speed and intrinsic domain- and language-independence of TRW's **hardware data pattern matcher**, with the organizational capabilities and inferential power of ISI's **semantic classification system**. Although both FDF and Loom have been used in numerous other applications, they have never been combined, and the combination offers a natural opportunity for the two to leverage off each other's strengths.

The heart of the system is TRW's *Fast Data Finder* (FDF). Empirical use has found that, though very good at document recall (that is, the FDF usually finds close to all the relevant material), the retrieval is weak in precision (that is, numerous spurious documents are retrieved as well), unless a specialist with domain expertise constru s the queries. This proposal addresses that problem in two ways: by enabling the system to construct its own queries out of a set of seed documents, and by providing sophisticated yet easy-to-use semantic modeling methods for aiding nonspecialists in constructing powerful queries.

There are three major claims:

1. The system is able to understand and reason about queries and documents in order to **construct precise queries and perform document routing** automatically. It uses a form of shallow semantics called a Domain Model and classificatory reasoning capabilities to disambiguate and refine user-constructed queries and to determine the clustering of documents by related topics.

2. The proposed system is **not language-specific**, in that it depends in no way on syntax or morphology. Extending its performance to a new language is essentially a matter of providing a lexicon of words of the new language and linking them up to the central English lexicon.

3. The system is **easy to use**, even by non-specialists, **and fast**. It can either construct queries automatically, upon being given a set of seed documents, or it can accept users' queries, which they form simply by mouse clicking nodes in the graphic display of the Domain Model. The core search engine is embodied in a very fast hardware pattern matcher.

The three sites collaborating on the proposed system are the Information Sciences Institute of the University of Southern California (ISI), TRW Inc., and SYSTRAN Inc. The first two institutions are in the Los Angeles area, the last near San Diego. The prototype will be constructed at ISI with help from TRW and using Japanese expertise from SYSTRAN (years 1 and 2). The conversion from prototype to large-scale system will take place at TRW, with help from ISI and SYSTRAN (years 3 and 4).

2.4.1 The Idea

A major problem facing current Information Retrieval systems is the difficulty of performing well both on recall — getting *all* the relevant information — and precision — getting *just* the relevant information — in complex domains with densely written texts. Existing fast brute-force techniques perform extremely well on recall, but suffer from finding too much irrelevant material when the text complexity is high or when the user is not an expert in query formulation. Although these methods can be made to provide essentially 100% recall, the cost is a lack of precision.

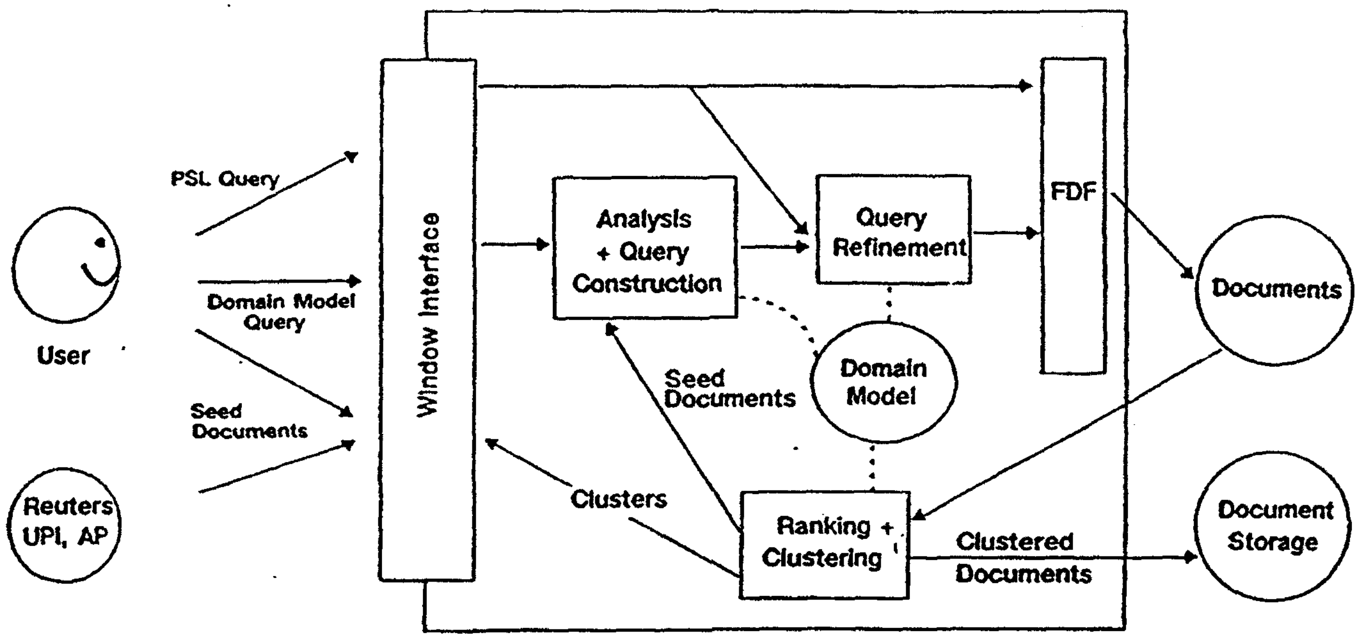
There is only one answer: improving precision requires greater expertise in formulating queries. Traditionally, this means that experts in information retrieval have to be employed — and even they spend a lot of time sifting through retrieved material, looking for relevant documents.

This proposal outlines a novel method that reduces the need for such expertise in retrieval query construction. The basic idea is to provide the system with additional knowledge and reasoning mechanisms so that it can help the user with query construction and document routing, or can perform these tasks alone when given one or more seed documents to start with. The proposed system will be able to operate at various levels of automation, allowing the user to specify which part(s) of the problem he or she wishes to perform manually.

The additional mechanisms that enable query refinement and construction and document routing are semantics-based classification techniques. However, since full semantic processing is not a practical reality at this time, we propose to use a limited form of semantics we call *shallow semantics.* Shallow semantics takes the form of Domain Models represented in ISI's knowledge representation system Loom [MacGregor 88, MacGregor & Bates 87]. As we will show, the addition of Domain Models enables the system to 'understand' the queries constructed by humans and also to form its own queries automatically while scanning documents. By 'understanding' the queries, the system is, among other things, able to:

- find ambiguous query terms (terms with multiple meanings, that may cause undesired documents to be retrieved), and disambiguate them;
- find additional terms for inclusion into the query, in order to make it more precise;
- accept terms and automatically form queries out of them, optimally combining them using the query language's Boolean operators *and, or* and *not*;
- recognize the semantic interrelationships of words from retrieved documents, thereby clustering them for classification and routing.

Using these models, the user can easily identify concepts relevant to his/her problem, select appropriate ones, and have the system automatically construct accurate and precise queries from them. On receiving the retrieved documents (sorted by topic cluster and ranked, as described



below), the user can specify either which documents he/she wants to read, or can select some subset of them as seed documents for further queries, or can ask for the most pertinent (that is, frequently-occurring) terms in order to enter t m into the Domain Model for future use (by the user and others). Alternatively, the user can specify a set of seed documents and have the system retrieve all similar documents and route them to appropriate destinations.

2.4.2 Using the System: A Flowchart

The proposed system can be used in two basic modes: interactively, with a human user, or in automatic mode, operating of seed documents which were chosen for their applicability to the desired topic of investigation. The overall system architecture and flow of information is shown in Figure 1.

During 1990, a proposal was written and submitted to Darpa to perform two to four years' work building a prototype Information Retrieval system. In collaboration with TRW Inc. and Systran, the proposal described linking together the following existing technology: Loom-based semantic modeling and classification (ISI), a hardware chip that performs very rapid data access and matching (TRW), and Japanese lexicons (Systran). Unfortunately, this work was not funded. We continue to believe however in its essential workability and value and will continue to search for funding opportunities.

2.5 Machine Translation

During 1989, Dr. Eduard Hovy made a presentation at the Darpa workshop and two drafts of a white paper suggesting the establishment of a new nationwide Machine Translation program were written by Dr. Eduard Hovy, the latter in collaboration with Drs. Sergei Nirenburg, Jaime Carbonell (both from CMU), and Yorick Wilks (from CRL, New Mexico). This discussion was continued during 1990, eventually giving rise to a new collaborative project.

In collaboration with Drs. Sergei Nirenburg and Jaime Carbonell from CMU and Dr. Yorick Wilks from CRL, a white paper was prepared for presentation to Darpa in mid-1989. This white paper proposes a new nationwide research program in the automatic translation of human languages by computer (known as Machine Translation (MT) or, when performed with human aid, as Machine-Aided Translation (MAT)).

The possibility of using computers to perform the translation of documents in various languages was one of the earliest goals of Natural Language Processing and, indeed, one of the earliest of Artificial Intelligence. In the typical approach taken at the time, a parser program was equipped with a grammar and lexicon of the source language and a generator program with a grammar and lexicon of the target language, and the remainder consisted of a set of syntax- and lexicon-correspondence rules. These approaches were soon proved naive by translations such as the now-famous "the vodka is strong but the meat is rotten" from "the spirit is willing but the flesh is weak". It was apparent that semantic information had somehow to be taken seriously (at least to the point of knowing that "spirit" may indeed be "vodka", but not when used as an active agent who can be "willing").

Since the early 60's, Machine Translation (MT) as a field of inquiry has largely lain dormant in the U.S., with the exception of a few large projects such as a project at the University of Texas (Austin) and a few smaller projects such as Lytinen's thesis at Yale University [Lytinen 84]. In recent years, however, especially under the impetus of Japanese and European efforts at addressing the problem, U.S. interest in MT research has been on the increase.

The principal reason for the increase is the ongoing development of tools and techniques that enable us to perform certain tasks with more thoroughness and success than was possible earlier (see, for example, [Carbonell et al. 81, Carbonell & Tomita 87, Nirenburg 87], [Arnold 86, Nakamura et al. 88, Laubsch et al. 84, Amano 86]). Not only has there been a steady growth of the capabilities of parsers and generators, the coverage of grammars, and the power and sophistication of knowledge representation techniques, but two recent developments have the nature of breakthroughs and will greatly enhance future MT systems: the incorporation of disjunction in KL-ONE-like representation systems, and the development of general-purpose language-based taxonomical ontologies of representation. This means that we are now in a better position to estimate the complexity of the problem and to pinpoint what we can hope to do and what remains beyond our grasp.

The field has grown wiser since the 60's: the newer MT projects are all less ambitious in scope than the early ones. Though nobody today would promise to deliver a system that performs perfect translation in even a relatively restricted domain, researchers feel comfortable about proposing systems that perform the first pass of a translation, producing a rough copy of the text in the target language, which would then be edited for stylistic smoothness and fluent cadence by a human editor. Since such systems significantly reduce the problems and costs of translation, they are in high demand in industry and industrial research throughout the world. For example, the following passage is from the invitation to an international seminar on MT organized by IBM, to be held in Munich, West Germany, in August 1989:

> There is a growing need for translation (estimated at 15-25 percent per annum) in commerce, science, governments, and international organizations. This is due to increased international cooperation and competition, an ever-growing volume of text to be communicated, often in multiple languages, world-wide electronic communication, and more emphasis in countries on the use of national language in documents and systems. The opening of the European market in 1992 will add significantly to these factors.
>
> At the same time, automated machine translation of natural language is reaching the stage where it can deliver significant cost savings in translation production, and vastly increase the scope of information retrieval, although fully automated high-quality translation is technically not feasible today and in the near future.
>
> [H. Lehmann and P. Newman, IBM Scientific Centers in Heidelberg and Los Angeles, 1989.]

2.5.1 The Components of an MT System

In order to build an MT system, the following program modules or components are needed:

- A Parser
- A Generator
- Grammars for each language
- Two lexicons
- A semantic Knowledge Base
- Interlanguage Translation Rules (in systems without an interlingua)

Parser: Sentences of the source text are parsed into some internal form by the parser. In almost all current MT systems, the internal form represents both syntactic and semantic aspects of the input.

Interlanguage Translation Rules: Many MT systems contain a set of rules that transform certain aspects of the internal representation of the input to make them conform to the requirements of the target language. Such MT systems are known as *transfer-based*. An alternative approach is to build MT systems without transfer rules, using a single intermediate representation form called an *interlingua*; the generality and power of such systems depends on the expressiveness of the interlingua used.

Generator: The (modified) internal representation of the input is generated as sentence(s) of the target language by the generator. The output must express the semantic content of the internal form, and if possible should use syntactic forms equivalent to those present in the input.

Grammars: In some systems, the grammars (syntactic information) are intrinsic parts of the parser and generator; in others, the grammars can be separated from the procedural mechanism. In *bidirectional* systems, the parser and generator use the same grammar to analyze and produce each language. Such systems are desirable because they do not duplicate syntactic information and are therefore more maintainable. True bidirectional grammars have proven hard to build, not least because existing knowledge representation formalisms do not provide some capabilities (such as inference over disjunction) that facilitate parsing and generation.

Semantic Knowledge Base: All sophisticated MT systems make heavy use of a knowledge base (representing underlying semantic information) containing the ontology of the application domain: the entities and their possible interrelationships. Among other uses, the parser requires these entities to perform semantic disambiguation and the generator uses them to determine acceptable paraphrases where exact 'literal' formulations are not possible.

Lexicons: All MT systems require a lexicon for the source language and one for the target language. In simple systems, corresponding entries in the two lexicons are directly linked to each other; in more sophisticated systems, lexicon entries are either accessed by entities represented in the knowledge base, or are indexed by characteristic collections of features (as built up by the parser).

2.5.2 The Use of Penman in a Machine-Aided Translation System

The three cornerstones of an MT system are the parser, the generator, and the knowledge representation system. The Penman project of USC/ISI embodies very sophisticated parsing and generation capabilities, as well as a general-purpose representation ontology that is highly suited for natural language processing of all kinds. USC/ISI is poised to investigate the questions of MT and to produce a system that performs a limited form of MT known as Machine-Aided Translation (MAT), in which the system performs a first pass of the translation and then a human editor performs the second pass.

Generation with Penman Penman is a natural language sentence generation program developed at USC/ISI. It provides computational technology for generating English sentences and paragraphs, starting with input specifications of a non-linguistic kind. The culmination of a continuous research effort since 1978, Penman contains one of the largest computational grammars of English in the world, and has been distributed to approximately 25 locations. Penman's structure and use is described in detail in the Penman Primer, User Guide, and Manual [Penman 88].

Penman consists of a number of components. Nigel, the English grammar, is the heart of the system. Based on the theory of systemic linguistics (a theory of language and communication developed by Halliday and others [Halliday 85, Halliday 73, Halliday 67, Halliday 66], and used in various other AI applications such as SHRDLU [Winograd 72], [Davey 79], [Patten 88]), Nigel is a network of over 600 nodes, each node representing a single minimal grammatical alternation.

Guided by its input, communicative goals, and other settings, Penman traverses the network, selecting features at each node, until it has assembled enough features — typically, about 100 — to fully specify a sentence. Using these features to control the selection, ordering, and inflection of words, it then generates the English sentence. Nigel is described in [Matthiessen 84, Matthiessen 87a, Matthiessen 87b, Mann & Matthiessen 83, Mann 83, Mann 83].

To simplify the control of Penman, the system also contains a number of auxiliary information resources, such as a lexicon of words (containing word definitions, inflectional forms, etc.) and a taxonomic model of the world. This taxonomy, called the *Upper Model*, is represented in the Loom knowledge representation system [MacGregor & Bates 87], and is based on the distinctions made in English. For example, since objects are treated differently in English than actions, objects and actions are defined as different classes in the model.

Knowledge Representation using Loom The knowledge representation language Loom [MacGregor & Bates 87] is being developed at ISI as a successor to NIKL in the KL-ONE tradition. Loom is already functional to the point of being distributed to the computational community. Currently, Penman's Upper Model is represented in Loom, and Loom is distributed together with Penman and is the suggested system for the construction of the application domain's Domain Model (which contains a taxonomy of the entities and relationships present in the domain). Current Loom development includes incorporating the treatment of disjunction and negation to enable proper inference over disjoint cases.

Parsing Over the past two years, under AFOSR funding, a member of the Penman project has adapted a parser, built as part of his dissertation work, to use Penman's grammar Nigel [Kasper 88b]. The first version of this parser was built by extending PATR-II [Shieber 84], a general unification-based system. It demonstrated that a general parsing capability could be developed for systemic grammars that are expressed in a declarative notation. The next version of this parser will explore how semantic information can be incorporated into the parsing process. Having incorporated the ability to perform inference over disjunction in Loom, both semantic information (as captured in the Upper and Domain Models) and syntactic information (Nigel, represented in Loom) will be accessible by the parser in a straightforward and homogeneous way. Furthermore, to aid the parsing process, Loom's classifier will be available to the parser as a useful and fast inference engine, and will take the place of the unification mechanism used previously. See the previous Quarterly Technical Report for more details.

The experiment of integrating a parser and generator with both semantic and syntactic knowledge represented in a KL-ONE-like representation system has never been carried out before. The imminent development of this capability is an exciting new breakthrough on the way to full bidirectional MT.

Current Experience in MT with Penman Given Penman's grammatical coverage and facility of use, it is not surprising that interest has been expressed in using the program in an MT system. This interest has expressed itself as follows:

- **Collaboration with EUROTRA:** The EUROTRA project is a multinational Machine Translation endeavor funded by the European Community (EC), with the task of developing programs to translate technical documents between the languages of the EC countries. Members of the German branch of the EUROTRA project spent two months visiting ISI in mid-1988 to examine Penman and decided to use it in pursuing their research in Germany. The German branch of EUROTRA resides at the IAI, a research institute associated with the University of the Saarland at Saarbrücken, West Germany. This collaboration is continuing: in order to participate in the development of techniques and information sources at the IAI, a member of the Penman Project spent 2 months in Saarbrücken early in 1989. Further collaboration with EUROTRA is expected throughout 1990.

- **Papers recently published by the group:** So far, this collaboration has resulted in two published papers in refereed conferences (the former being one of the major NLP conferences in Europe, namely the European ACL): [Bateman et al. 89a] and [Bateman et al. 89b]. The papers describe the initial results of the experiment of linking the parser developed at the IAI with Penman, operating to the requirements imposed by the overall EUROTRA project.

- **Invitation to an international MT workshop:** Given our relatively recent interest in MT, it was an unexpected honor for a member of the Penman team to be invited to speak at an international seminar on MT to be held in Munich, West Germany, in August 1989. This seminar is organized by IBM, whose interest in MT is substantial (as expressed in the large number of separate IBM-internal MT projects in various countries).

Work Required to Produce a Penman-Based MAT System As can be seen, USC/ISI already has in place most of the components required for a sophisticated MAT system. What remains to be done is the following:

- **Development of a target-language grammar.** To date, Penman has generated only English. Rather than develop the second grammar ourselves, we are making use of the offer we recently received from the KOMET project of the GMD in Germany to collaborate with them, where in return for providing them with Penman they will provide us with a grammar of German, phased over three years.

- **Adaptation of the parser to use Loom and other grammars.** The parser has already been demonstrated at ISI using Penman's grammar. Its adaptation to Loom is currently under progress. The same parsing method will be applied to a German grammar, which is feasible since the German grammar uses the same framework of Systemic-Functional Linguistics as Penman does.

- **Representation of the application domain terms.** A suitable application domain with easily available, preferably already computer-internal data must be selected and made accessible to Penman and the parser. This requires representing the objects and relations of the domain in a Domain Model, the definition of lexical items, and the definition of certain access functions that link the system and the data. This is not a task of large complexity.

- **Integration of all the components.** The use of the Upper and Domain Model terms, in conjunction with some taxonomic transfer rules, as a type of linguistically based interlingua, and the use of the English and German grammars in bidirectional fashion enforces a coherence among the various components in the abovementioned MAT system based on Penman. A certain amount of remaining tailoring is inevitable, especially in areas where German is more complex than English, such as lexical morphology (verb and noun endings, for example), to ensure that all the existing components can make use of all the new features.

The work that needs to be done at USC/ISI, as outlined above, requires the efforts of four researchers:

- A grammar specialist (full-time): This person will be responsible for representing Penman's grammar in Loom, for ensuring that the grammar can support the new domain, for defining the Domain Model and subordinating it to the Upper Model, and for performing the generation of the domain texts.

- A parser specialist (full-time): This person will be responsible for completing additions to Loom's capabilities, for integrating the parser with Loom, co-responsible for representing the domain, and for performing the parsing of the domain texts.

- A generator specialist (half-time): This person will be responsible for ensuring that the German grammar conforms to the requirements of Penman and for the embedding of the grammar in Penman, including the extension of Penman's capabilities to deal with the more complex morphology of German, co-responsible for representing the domain, and for assisting with the generation of the domain texts.

- A text specialist (half-time): This person will be responsible for developing the representational terms required for parsing multisentence texts (as opposed to isolated sentences), for incorporating these terms into the grammars in such a way that they can be used by the parser and the generator, and co-responsible for representing the domain.

The benefit to DARPA and the Natural Language Computational community is clear. For relatively little expense, a major new MT effort will come into being in the next two years. Much leverage will be gained from the collaboration with the GMD, and the existing generation and parsing capabilities of the Penman project will be used to maximum effect.

Penman's Suitability for MAT Penman is well suited to form part of a Machine-Aided Translation system due to a number of factors. First, its clean formulation, wide linguistic scope, and ease of use makes it a very good prospect in terms of utility and maintainability.

Second, Penman is especially well suited to MAT because its generation procedure is controlled in part by a general ontological model of the entities and relationships in the world (as distinguished in English) called the Upper Model. These entities and their interrelationships are organized into a property-inheritance network, under which the entities and relationships of the application domain are classified. During the traversal of Penman's grammar while building up a sentence, Penman directs its taxonomic queries to the Upper Model, whose organization is imparted to the propositional content that needs to be expressed in the text. The Upper Model thus serves as the basis of the underlying knowledge base, and has in fact been so used (with some extensions) in collaborations with various other projects (in fact, the Upper Model has found use as a taxonomic representational device in various applications, including a number that do not include Penman at all). In conjunction with the domain-specific entities in the Domain Model, the Upper Model entities embody a set of terms that are rich enough to capture the nuances of meaning present in the domain and have enough organizational structure to support the parsing and generation of language. Thus the Upper and Domain Models contain linguistic generalizations of a semantic nature, many of which remain constant over different languages (especially in the more abstract reaches of the Upper Model, taxonomizing the world into objects, qualities, and processes such as actions, events, and relations), there is very little reason to expect any differences between the standard Westernized languages of Europe and Japan. Thus to the extent that English shares with the other languages an underlying ontology of the world, the Upper and Domain Models can act as a type of interlingua in an MT system, where differences are taken care of by transfer rules of the normal type. This linguistically motivated semi-interlingua to capture generalizations is preferable to the lexically based and pure transfer approaches, both of which involve large numbers of special-purpose rules.

Third, work is currently under way in developing a parser companion to Penman. A member of the Penman group, a specialist in parsing, has been performing work to extend the capabilities of the knowledge representation language Loom. The ability to handle disjunction should enable, by the end of this year, the representation of Penman's grammar in Loom. A previously developed parser will then be adapted to parse English sentences using Loom's automatic concept classifier to classify the input with respect to the grammar as well as to Penman's Upper Model. Since the grammar is functional and semantically oriented, this parsing capability should result in a truly bidirectional grammar. Bidirectional grammars have the obvious advantage that the same grammar which is used for generation is immediately available for parsing. Any enhancements to the basic grammar traversal mechanisms and supporting knowledge sources immediately benefit all versions of Penman, while development of grammars for different languages can proceed independently.

Fourth, Penman is highly modularized, enabling its grammar to be separated from its grammar traversal mechanism at will. This means that any other grammar built up according to the same underlying principles of Systemic Linguistics can be mounted within the Penman system with little or no special tailoring. Thus, at the cost of developing a grammar for a new language (or rewriting an existing grammar in the right format) a new language can be generated.

As a result of the last two factors, once the basic representational mechanism has been developed, the Penman project will be able to investigate MT with minimal overhead devoted to generation and parsing mechanisms, and with maximal attention given to the development of the grammars and the underlying semantic representations. In fact, there is current discussion with another research group in Germany, funded by the German government through the GMD (a country-wide institute for research in Mathematics and Computer Science with over 1,000 researchers), about the development by them of a German version of Penman in exactly this fashion. The resulting system would include a parser and generator with an English and a German grammar.

For these reasons, we believed that Penman is admirably suited for a full-scale investigation into Machine Translation. The 15 man-years of work on Penman and supporting subsystems have resulted in the major components of an MT system. What remains is the development of representational techniques to support the formulation of grammars in Loom, the linking of the parser to this grammar representation, and the development and incorporation of the grammar of some other language into this framework.

2.5.3 Statistical Work

During 1990, Dr. Kenneth Church visited the project from AT&T Bell Laboratories for one year. Dr. Church's work on the automatic construction of bilingual lexicons is in strong support of the current MT effort; only by having the ability to easily and quickly gather large quantities of words can we realistically hope to build an MT system with wide enough coverage to prove interesting in the limited time we have.

In collaboration with Dr. Hovy, Dr. Church developed computer systems to achieve the alignment of sentences from multilingual parallel texts (in this case, a collection of French-German-English banking texts obtained from Switzerland and an extract of the Canadian French-English parliamentary Hansard). He then developed statistical methods for correlating words in these sentences so as to identify the cross-language pairs. This work will be reported in next years' conferences.

In early 1991, Dr. Ken Church continued his work on the construction of bilingual lexicons based on the Canadian Parliamentary Hansard. A paper describing this work was presented at the Darpa Speech and Natural Language Workshop in Asilomar, CA, in February.

2.5.4 Progress of MT Proposal

In 1990, a proposal was written and submitted to Darpa to perform three to five years' work building a prototype Machine-Aided Translation system. Under the proposal, a standalone system could be built at USC/ISI, or the system could be incorporated with the efforts of CMU's Center for Machine Translation and New Mexico State's Computing Research Laboratory; these three sites' proposed work has been designed to fit together. After submitting the proposal, we continued work on the essential preparation for it, should the proposal be successful. This involved extending the prototype Penman parser, extending the capabilities of the Loom Knowledge Representation language (which is of central importance to the parser), and transferring Penman's extensive English grammar into a Loom representation.

In mid-1990 we received word that our joint proposal with CMU and CRL was accepted. Therefore, one of the principal thrusts of the next three to five years' work will be the development of a machine-aided translation system based on Systemic Linguistics, using Penman, our parser, and the Upper Model as a general Interlingual / Transfer Structure. In this regard, we continued work on completing the parser and finding a new colleague to replace Dr. Robert Kasper. During the second half of 1990, we collected a large number of applications for the open position (approximately equally many from Europe and from the U.S., including two senior lecturers from highly-regarded British universities). We interviewed three candidates —

Dr. Peter Norvig from UC Berkeley, Dr. Yves Schabes from the University of Pennsylvania, and Dr. Mark Seligman (a graduate student on the point of finishing his thesis at UC Berkeley). Somewhat later we interviewed two other candidates, Mr. Mike Reape (a graduate student at the University of Edinburgh) and Dr. Chinatsu Aone from MCC. We decided however not to employ anyone at the time.

In related work in early 1991, Drs. Hovy and Church wrote a survey paper on the various approaches and metrics proposed for MT systems, in order to define future MT work more precisely and to develop it toward maximum effectiveness.

In March 1991, the first cross-project meeting of representatives from CMU, CRL, and ISI was held in Las Cruces, NM. The new system was named PANGLOSS. Discussions focussed on the domain to be selected, the ontology of representation, evaluations, and site responsibilities.

3 Significant Hardware Developments

None.

4 Equipment

In December 1990, a new Sun SPARCstation 1+, with accompanying large disk storage, was acquired for the use of Dr. Ken Church.

In March 1991, a stripped-down Sun SPARCstation 1+ was acquired on a special deal from Sun, on trading in a small old Sun 3.

5 Key Personnel

In March 1987, Dr. Eduard Hovy joined the project. Dr. Hovy had just completed a Ph.D. in language generation at Yale University, and joined the project with the intent of working on multisentential planning.

In September 1987, Dr. John Bateman joined the project. After graduating from the University of Edinburgh, Dr. Bateman had spent two years doing postdoctoral work at the University of Kyoto. His specialties are discourse and grammar from a systemic point of view.

In September 1987, Ms. Lynn Poulton, a graduate student in Linguistics at the University of Sydney who had been with the project for two years, left to continue her studies.

In November 1987, Mr. Tom Galloway, a programmer who had been with the project for about a year, left to work at the University of Geneva.

During late 1988 to early 1989, funding for the project was greatly reduced. In order to make ends meet, Dr. William Mann, project leader, went on partial retirement; several project members left the project (including Mr. Robert Albano, Mr. Christian Matthiessen, and Ms. Lynn Poulton); and the following project members worked on other projects temporarily:

- In 1989, Dr. John Bateman spent three months at the IPSI institute in Darmstadt, Germany.
- In 1989, Dr. Eduard Hovy spent three months working in the LILOG project at IBM in Stuttgart, Germany.
- In 1989, Dr. Robert Kasper spent three months teaching at the Summer Institute of Linguistics in North Dakota.

Throughout 1989, due to shortage of funding, Dr. John Bateman spent most of the year working at IPSI, our West German collaborators in Darmstadt. Even though in constant electronic mail contact with ISI, his leaving left a gap in our ability to handle efficiently queries and extensions to the grammar and the generator. For that reason, we employed a graduate student from the University of Sydney, Australia, who was pursuing a Ph.D. in Systemic Linguistics. The student, Mr. Mick O'Donnell, has some computational experience and was highly recommended.

The funding shortage continued throughout 1990 (and still continues today). During 1990, Dr. Bateman continued to spend most his time working at IPSI. He was appointed leader of the generation project at IPSI in order better to organize the research they are conducting. Currently, they are building a German grammar and are augmenting Penman's Upper Model.

In June 1990, Dr. Robert Kasper left ISI to join the Linguistics Department of the Ohio State University. Although his reasons for leaving were private, his departure eased the financial straits of the project somewhat. However, he was sorely missed. Dr. Kasper continues his work on parsing, and will collaborate with the Penman project throughout the next 18 months. A plan of consulting of approximately 10 days every two months, in addition to communicating via email and telephone, has been agreed upon.

In September 1990, Dr. Ken Church joined the Penman group at ISI for a year-long sabbatical. He immediately began working on the automatic construction of bilingual lexicons.

At the end of 1990, Mr. Mick O'Donnell returned to Sydney University to complete his Ph.D. thesis. He had spent a little over a year at ISI, and during this time proved himself a very useful and productive team member. His major contributions were to standardize, expand, and simplify the use of Penman on the Macintosh, and to help develop the parser (which is the focus of his own Ph.D. research as well).

During 1991, due to continued shortage of funding, Dr. Bateman spent his time leading the KOMET project (sister to the Penman project) at the IPSI institute in Germany.

In April 1991, Ms. Elisabeth Maier, a graduate student at IPSI who is performing research on text planning, spent six weeks helping with the construction of the new text planner.

6 Trips and Conferences

During the period of this contract, project members attended the following conferences and meetings:

- Dr. Norm Sondheimer was the chair of the organizing committee of the Applied ACL conference, Austin, TX, February 1987.
- Five project members attended the third TINLAP workshop, New Mexico, February 1987. Drs. Mann and Sondheimer were both panelists.
- Four project members attended the DARPA workshop in May 1987, during which the joint BBN-ISI natural language dialogue system JANUS was demonstrated.
- Four project members attended the Association of Computational Linguistics (ACL) conference in July 1987. One paper from the project was presented. Dr. Mann, the then president of the Association, gave the banquet address.
- Dr. Bill Mann and Mr. Christian Matthiessen attended the Systemics Workshop in Sydney, Australia, where they each delivered a paper. Dr. Bateman also attended the workshop, although he had not yet joined the project.
- Three members of the project attended AAAI in August 1987. One paper was presented.
- Three project members attended the DARPA evaluation workshop in Palo Alto in November 1987, and presented two papers.
- Dr. Hovy attended the DARPA meeting in Dallas, Texas, November 1987.
- Dr. Hovy visited the University of Delaware in Newark, Delaware, in November 1987, to collaborate with Prof. McCoy on work which extends Penman's text planning capabilities; to give a colloquium, a seminar, and a lecture; and to assess the possibility of making Penman available to NLP researchers at the university.
- Dr. Hovy attended the NLP Evaluation workshop in Philadelphia in December 1987. He was on the program committee and was a session chair.
- Five project members attended the Association of Computational Linguistics (ACL) conference in Buffalo, NY, June 1988. Four papers project were presented.
- Four project members attended the International Conference on Systemic Linguistics, East Lansing, Michigan, July 1988. Five papers and a tutorial were presented.
- Two project members attended the AAAI Conference in Seattle, WA. One paper and one workshop presentation were delivered.
- Dr. Eduard Hovy presented a paper outlining a possible new program in Machine Translation at the Darpa Speech and Natural Language workshop on Cape Cod, March 1989.
- Dr. Robert Kasper presented a paper on the new developments and style of parsing, as outlined in the Technical Report, at the International Parsing Workshop (of which he was an organizing committee member) in Pittsburgh in October 1989.
- Dr. Robert Kasper was invited to spend two weeks at IPSI in November 1989, in order to install his parser and describe the next few months' of parsing research that he will do in collaboration with their KOMET project.

- Dr. Eduard Hovy visited the Universities of Waterloo, Toronto, and Carnegie Mellon University in January 1990, where he gave colloquia and interacted with graduate students. At CMU he also worked on the Machine Translation initiative with Drs. Jaime Carbonell and Sergei Nirenburg.
- Dr. Eduard Hovy attended the Workshop on Knowledge Representation Standards in Santa Barbara in March 1990, in the place of Dr. Bill Swartout of ISI.
- Dr. Eduard Hovy attended the AAAI Symposium on Human-Computer Interfaces at Stanford University in March 1990, where a paper he co-authored with Dr. Yigal Arens from ISI was presented by Dr. Arens.
- Drs. Bateman and Hovy and Mr. Mick O'Donnell attended the International Workshop on Language Generation in Pittsburgh in June 1990. Two papers by Dr. Bateman and one by Dr. Hovy were presented. The same three people attended the Annual Meeting of the Association for Computational Linguistics directly afterward. Mr. O'Donnell demonstrated the MacIntosh-II version of Penman at the conference.
- On vacation in Italy in April 1990, Dr. Hovy visited the Natural Language Project at the Instituto per la Ricerca Scientifica e Technologica in Trento, where he gave a colloquium and interacted with project members for a day.
- Dr. Eduard Hovy presented the following talks: A talk on the future possibilities of language generation at the AI Systems in Government conference in May 1990; an evening talk on Penman to the Los Angeles chapter chapter of SIGART in May 1990; a talk on parsing using classification-based techniques at the Darpa Speech and Natural Language Workshop in June 1990; a talk on pragmatics at the Rocky Mountain AI Conference on Pragmatics in Las Cruces in June 1990.
- Dr. Hovy attended the annual AAAI conference in Boston in July 1990, where he presented two papers in workshops: one on Evaluation of Language Generation systems and one on the interactive communication of multiple agents in a world simulation system. He also attended the Cognitive Science Society annual meeting at MIT, where he presented a paper (co-authored with Dr. Yigal Arens of ISI) on multimedia (natural language, graphs, tables, etc.) communication.
- In September 1990, Dr. Hovy attended the KBSA workshop in Syracuse, NY, where he presented a paper (also co-authored with Dr. Yigal Arens of ISI) on the relation between multisentence text planning and multimedia communication planning. He also appeared on a panel at the workshop, discussing the role of general language-based knowledge in KBSA-like systems.
- Mr. O'Donnell spent two months in Europe, one of them at IPSI (our German partner) and the other visiting various universities and research sites in Canada, England, and Wales. He installed Penman in Toronto and in Cardiff, and helped the British Telecom research lab with problems they had been having using Penman in their own research.
- On invitation to participate in a week-long colloquium on NL, Dr. Hovy spent October 1990 in Germany, working in addition one week at IPSI with the Penman team there, one

week at the IBM Research center on NL in Stuttgart (where in 1989 he had worked for two months designing a text planner for them), and a week at the FAW institute in Ulm, planning out a new multimodal text/graphics planning system in collaboration with Dr. Dietmar Rösner.

- Dr. Ken Church attended a conference on lexical issues in Montreal, Canada, on whose organizing committee he had served. He presented a paper there and met with MT researchers in Montreal.
- In March 1991, Drs. Hovy and Church attended the Darpa Speech and Natural Language Workshop in Asilomar CA, where Dr. Church presented a paper describing the multilingual lexicon construction work he has been doing.
- Dr. Hovy attended the first MT planning meeting at the CRL in New Mexico to discuss the new MT project with representatives from CMU and CRL.
- Dr. Ken Church spent a month in England and Switzerland, visiting the lexicographers at the Oxford Dictionary and elsewhere and MT researchers in Dr. Maghie King's group in Geneva, Switzerland.
- Dr. Hovy attended the IEEE Conference on AI and Applications in Miami FL, in February 1991, where he appeared on a panel to argue for language-based generalization hierarchies as the best choice for interdomain knowledge representation ontology primitives.
- On invitation to deliver the keynote address at the 3rd European Workshop on Language Generation, Dr. Hovy attended the workshop in Austria for a week in March 1991. He also appeared on a panel at that time.

7 Project-Related Visitors

- **Longer visits during 1988:**
  - For three months during the summer, three researchers from the University of Saarbrücken, W. Germany, visited the project. Dr. Erich Steiner, Mr. Joerg Schütz, and Ms. Elke Teich are members of Eurotra-D, the German team of the European MT project. They have developed parsing technology based on the same functional principles from which Penman is derived, and were visiting to investigate the possibility of using Penman as the generation component of their system. The visit was successful; within short order they were able to perform some translations. They have started the process of acquiring Penman for their own use.
- **Brief visits in 1989:**
  - March 20: Prof. Graeme Hirst, University of Toronto
  - June 30: Dr. Martin Emele, ATR, Kyoto (and Uni Stuttgart)
  - June 30: Ms. Penni Sibun, UMass, Amherst
  - July 1 – 10: Ms. Chrysanne DiMarco, Uni Toronto

- July 10: Dr. David Farwell, CRL at NMSU, Las Cruces
- August 9: Prof. Erich Neuhold, IPSI of GMD, Darmstadt
- Sepember 6: Dr. Kiyishi Kogure, ATR, Kyoto
- November 22: Prof. Dan Moldovan, USC, LA
- December 15 – 30: Dr. Christian Matthiessen, University of Sydney

- **Longer visits during 1990:**
  - Sepember 4 – August 31, 1991: Dr. Kenneth Church, AT&T Bell Labs, NJ. Dr. Church, sponsored partly by DARPA and partly by AT&T, conducted research in the statistical processing of large corpora of text, and performed initial surveys of the area of Machine Translation with members of the Penman project.
  - Sepember 17 – November 15, 1991: Prof. Julia Lavid, University Complutense of Madrid, Spain. Prof. Lavid, a text linguist in the Department of English Philology, worked closely with Penman and EES project members in the construction of a new text planner and the development of theories about theme and focus.
  - Sepember 17 – March 15, 1991: Ms. Mira Vossers, U of Nijmegen, Nijmegen, The Netherlands. Ms. Vossers, a graduate student, wrote a M.S. thesis under the supervision of Drs. Hovy and Arens in the area of multimedia communication.
  - November 1 – February 20, 1991: Mr. Giuseppe Carenini, IRST, Trento, Italy. Mr. Carenini worked with the text planning group on the construction of a new text planner.

- **Brief visits in 1990:**
  - January 4: Messrs. Isao Kawashima, Fumihiko Obashi, NTT, Japan
  - January 15 – 15: Mr. Mishu Koñyves-Toth, IPSI, Darmstadt
  - January 19: Dr. Ken Church, AT&T Bell Labs, NJ
  - January 22: Dr. Randy Sharp, IAI, Saarbruecken
  - January 30: Dr. Barry Boehm and party, DARPA, Washington DC
  - February 14: Dr. Ikuo Keshi, Sharp Corporation, Nara, Japan
  - March 9: Prof. Igor Mel'čuk, University of Montreal, Montreal
  - March 21: Dr. David Miller, JPL NASA, Pasadena, CA
  - March 28: Ms. Margaret Sarner, Uni Delaware, Newark, DE
  - March 30: Prof. Joseph Bates, CMU, Pittsburgh, PA
  - June 10 – 14: Dr. Dietmar Rösner, FAW, Ulm, West Germany
  - June 11 – 15: Dr. Donia Scott, Philips Research Labs, Surrey, England
  - June 21: Prof. Gerard Kempen, Max Planck Institute, Nijmegen, The Netherlands
  - July 16: Dr. Ueda, ATR, Japan
  - August 14: Prof. Peter Fries, Michigan State U, MI
  - August 20: Ms. Jennifer Chu, U of Waterloo, Waterloo, Canada

- August 27 – 31: Prof. Chrysanne DiMarco, U of Waterloo, Waterloo, Canada
- November 2: Mr. Hercules Dalianis, University of Stockholm, Stockholm, Sweden
- November 12 – 16: Ms. Nadia Ben Hassine, U of Waterloo, Waterloo, Canada
- November 26: Prof. Michael Walsh, U of Sydney, Sydney, Australia
- November 26 – 27: Dr. Dmitris Karagiannis, FAW, U of Ulm, Ulm, Germany

• bf Longer visits during 1991:

- May 15 – August 31: Ms. Elisabeth Maier, IPSI Darmstadt, Germany. Ms. Maier, a graduate student and member of Penman's sister project in Germany, worked closely with Penman and EES project members in the construction of a new text planner.
- May 15 – August 24: Ms. Lynn Poulton, Rice University, Houston. Ms. Poulton, a former member of the Penman project and currently a graduate student, returned to the project for new grammar development.

• **Brief visits in 1991:**

- January 5 – February 28: Ms. Elisabeth Maier, IPSI Darmstadt, Germany
- March 17: Mr. John Mackin, FUJITSU, Kawasaki Japan
- April 5 – 30: Ms. Elke Teich, IPSI Darmstadt
- May 14: Dr. David McDonald, Content Technology, Boston
- May 16: Prof. Susanna Cumming, University of Colorado at Boulder

8 Selected Publications Funded by this Work

References

[Bateman et al. 89a] Bateman, J.A., Kasper, R.T., Schütz, J. and Steiner, E. A New View on the Process of Translation. In *Proceedings of the European ACL Conference*, Manchester, 1989.

[Bateman et al. 89b] Bateman, J.A., Kasper, R.T., Schütz, J. and Steiner, E. Interfacing an English Text Generator with a German MT Analysis. To be published as *Proceedings of the Gesellschaft für linguistische Datenverarbeitung*, Springer, 1989.

[Bateman 90] Bateman, J.A. Upper Modeling: A Level of Semantics for Natural Language Processing. Presented at the *Fifth International Workshop on Language Generation*, Pittsburgh, June 1990.

[Bateman & Paris 89a] Bateman, J.A. and Paris, C.L. Phrasing a Text in Terms the User can Understand. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI*, Detroit, MI, 1989.

[Bateman et al. 90] Bateman, J.A., Kasper, R.T., Moore, J.D., Whitney, R.A. A General Organization of Knowledge for Natural Language Processing: The Penman Upper Model. USC/ISI Technical Report, Marina del Rey, 1990.

[Hovy 87] Hovy, E.H. Generating natural language under pragmatic constraints. *Journal of Pragmatics* XI(6), pp. 689–719, 1987.

[Hovy 88a] Hovy, E.H. Planning Coherent Multisentential Text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June, 1988.

[Hovy 89] Hovy, E.H. New Possibilities in Machine Translation. *Proceedings of the DARPA Speech and Natural Language Workshop*, Harwichport, MA, May 1989.

[Hovy & McCoy 89] Hovy, E.H. and McCoy, K.F. Focusing your RST: A Step toward Generating Coherent Multisentential Text. *Proceedings of the 11th Cognitive Science Conference*, Ann Arbor, MI, Aug. 1989.

[Hovy 90a] Hovy, E.H. Pragmatics and Natural Language Generation. *Artificial Intelligence* 43(2) (153–198), May 1990. Also available as USC/Information Sciences Institute Research Report ISI/RS-89-233.

[Hovy 90b] Hovy, E.H. Parsimonious and Profligate Approaches to the Question of Discourse Structure Relations. Submitted to the *5th International Workshop on Text Generation*, Pittsburgh, June 1990.

[Hovy 90c] Hovy, E.H. Approaches to the Planning of Coherent Text. *Natural Language in Artificial Intelligence and Computational Linguistics*, C.L. Paris, W.R. Swartout, and W.C. Mann (eds). Boston, MA: Kluwer Academic Publishers, 1990. Also available as USC/Information Sciences Institute Research Report ISI/RS-89-245.

[Hovy 90d] Hovy, E.H. Unresolved Issues in Paragraph Planning. *Current Research in Natural Language Generation*, R. Dale, C. Mellish, and M. Zock (eds) (17–45). New York, NY: Academic Press, 1990.

[Hovy 90e] Hovy, E.H. A New Level of Natural Language Generation Technology: Capabilities and Possibilities. *Proceedings of the 5th AI Systems in Government Conference*, Washington, DC, July 1990.

[Kasper 87] Kasper, R.T. A Unification Method for Disjunctive Feature Descriptions. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Stanford, CA. July 1987. Also available as USC/Information Sciences Institute Reprint RS-87-187.

[Kasper 88a] Kasper, R.T. Conditional Descriptions in Functional Unification Grammar. In *Proceedings of the 26th Annual Conference of the Association for Computational Linguistics*, Buffalo, 1988.

[Kasper 88b] Kasper, RT.. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, 1988.

[Kasper 88c] Kasper, R.T. Systemic Grammar and Functional Unification Grammar. In *Systemic Functional Approaches to Discourse*, Benson, J. and Greaves, W. (eds), Norwood, NJ: Ablex (in press).

[Kasper 89] Kasper, R.T. A Flexible Interface for Linking Applications to Penman's Sentence Generator. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, Philadelphia, PA, February 1989.

[Kasper & Hovy 90] Kasper, R.T. and Hovy, E.H. Classification-Based Parsing using Integrated Semantic and Syntactic Knowledge. In *Proceedings of the 3rd Darpa Speech and Natural Language Workshop*, Pittsburgh, PA, June 1990.

[Maier & Hovy 91] Hovy, E.H. A Metafunctionally Motivated Taxonomy for Discourse Structure Relations. (Co-authored with E. Maier). *Proceedings of the 3rd European Workshop on Language Generation*, Innsbruck, Austria, March 1991.

[Mann 88] William Mann. Toward a Theory of Reading Between the Lines. Presented at the 14th International Systemics Workshop, Sydney, Australia, August 1988.

[Mann & Thompson 87a] William Mann and Sandra Thompson. Rhetorical Structure Theory: Description and Construction of Text Structures. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen, G. (ed), Kluwer Academic Publishers, 1987.

[Mann & Thompson 87b] William Mann and Sandra Thompson. Rhetorical Structure Theory: A Framework for the Analysis of Texts. In *IPRA Papers in Pragmatics* 1, pp. 79–105, 1987.

[Mann & Thompson 88a] Mann, W.C. and Thompson, S.A. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text* 8(3) pp. 243–281, 1988.

[Matthiessen 87a] Matthiessen, C.M.I.M. Notes on the Organization of the Environment of a Text Generation Grammar. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen, G. (ed), Nijhoff, 1987. Also available as USC/ISI Research Report RR-87-177, 1987.

[Matthiessen 87b] Matthiessen, C.M.I.M. Semantics for Systemic Grammar: The Chooser and Inquiry Framework. In *Systemic Perspectives on Discourse*, 1987. Also available as USC/ISI Research Report RR-87-189, 1987.

[Matthiessen & Mann 88] Christian Matthiessen and William Mann. Functions of Language in Two Frameworks. Presented at the 14th International Systemics Workshop, Sydney, Australia, August 1988.

[Penman 88] *The Penman Primer, User Guide, and Reference Manual.* Unpublished USC/ISI documentation, 1988.

[Rounds & Kasper 86] Rounds, W. and Kasper, R.T. A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, Cambridge, MA, June 1986.

[Thompson & Mann 87] Sandra Thompson and William Mann. Antithesis: A Study in Clause Combining and Discourse Structure. In *Language Topics: Essays in Honour of M.A.K. Halliday*, Steele, R. and Threadgold, T. (eds), Benjamins, 1987.

9 References

References

[Amano 86] Amano, S. The Toshiba Machine Translation System. In *Japan Computer Quarterly*, Vol. 64, 'Machine Translation — Threat or Tool', pp. 32-35, 1986.

[Arnold 86] Arnold, D. Eurotra: A European Perspective on MT. In *Proceedings of the IEEE*, Vol. 74, pp. 979-992, 1986.

[Bateman et al. 89a] Bateman, J.A., Kasper, R.T., Schütz, J. and Steiner, E. A New View on the Process of Translation. In *Proceedings of the European ACL Conference*, Manchester, 1989.

[Bateman et al. 89b] Bateman, J.A., Kasper, R.T., Schütz, J. and Steiner, E. Interfacing an English Text Generator with a German MT Analysis. To be published as *Proceedings of the Gesellschaft für linguistische Datenverarbeitung*, Springer, 1989.

[Bateman et al. 90] Bateman, J.A., Kasper, R.T., Moore, J.D., Whitney, R.A. A General Organization of Knowledge for Natural Language Processing: The Penman Upper Model. USC/ISI Technical Report, Marina del Rey, 1990.

[Bobrow & Webber 80] Bobrow, R.J. and Webber, B. Knowledge Representation for Syntactic/Semantic Processing. In *Proceedings of the 1st Conference on Artificial Intelligence (AAAI)*, Stanford, CA, August, 1980.

[Carbonell & Tomita 87] Carbonell, J.G. and Tomita, M. Knowledge-Based Machine Translation, the CMU Approach. In *Machine Translation: Theoretical and Methodological Issues*, Nirenburg, S. (ed), Cambridge University Press, Cambridge, 1987.

[Carbonell et al. 81] Carbonell, J.G., Cullingford, R.E. and Gershman, A.V. Steps towards Knowledge-Based Machine Translation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 3, pp. 376-392, 1981.

[Davey 79] Davey, A. *Discourse Production*. Edinburgh University Press, Edinburgh, 1979.

[Eisele & Doerre 88] Eisele, A. and Doerre, J. Unification of Disjunctive Feature Descriptions. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June 1988.

[Halliday 66] Halliday, M.A.K. Some Notes on 'Deep' Grammar. In *Journal of Linguistics*, Vol. 2:1, pp. 57-67, 1965.

[Halliday 67] Halliday, M.A.K. Notes on Transitivity and Theme in English. In *Journal of Linguistics*, Vol. 3:1, pp. 37-81, 1967.

[Halliday 73] Halliday, M.A.K. *Explorations in the Functions of Language*. Edward Arnold: London, 1973.

[Halliday 76] Halliday, M.A.K. *System and Function in Language*. Kress G., (ed.), Oxford University Press, 1976.

[Halliday 85] Halliday, M.A.K. *Introduction to Functional Grammar*. Edward Arnold Press: London, 1985.

[Karttunen 84] Karttunen, L. Features and Values. In *Proceedings of the 10th International Conference on Computational Linguistics: COLING 84*, Stanford, CA, July 1984.

[Kasper 87] Kasper, R.T. A Unification Method for Disjunctive Feature Descriptions. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Stanford, CA, July 1987. Also available as USC/Information Sciences Institute Reprint RS-87-187.

[Kasper 88a] Kasper, R.T. Conditional Descriptions in Functional Unification Grammar. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June 1988.

[Kasper 88b] Kasper, RT.. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, 1988.

[Kasper 88c] Kasper, R.T. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, August, 1988.

[Kasper 89a] Kasper, R.T. A Flexible Interface for Linking Applications to Penman's Sentence Generator. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, Philadelphia, PA, February 1989.

[Kasper & Hovy 90] Kasper, R.T. and Hovy, E.H. Classification-Based Parsing using Integrated Semantic and Syntactic Knowledge. In *Proceedings of the 3rd Darpa Speech and Natural Language Workshop*, Pittsburgh, PA, June 1990.

[Kay 85] Kay, M. Parsing in Functional Unification Grammar. In *Natural Language Parsing*, Dowty, D., Karttunen, L. and Zwicky, A. (eds). Cambridge University Press: Cambridge, England, 1985.

[Laubsch et al. 84] Laubsch, J., Rösner, D., Hanakata, K. and Lesniewski, A. Language Generation from Conceptual Structure: Synthesis of German in a Japanese/German MT Project. In *Proceedings of the COLING 84*, Stanford, 1984.

[Lytinen 84] Lytinen, S.L. *The Organization of Knowledge in a Multi-Lingual, Integrated Parser*. Ph.D. dissertation, Yale University Research Report # 340, 1984.

[MacGregor & Bates 87] MacGregor, R. and Bates, R. The Loom Knowledge Representation Language. In *Proceedings of the Knowledge-Based Systems Workshop*, St. Louis, 1987. Also available as USC/Information Sciences Institute Research Report RS-87-188, 1987.

[MacGregor 88] MacGregor, R. A Deductive Pattern Matcher. In *Proceedings of AAAI-88, The Sixth National Conference on Artificial Intelligence*, St. Paul, MN, August 1988.

[Mann 83] Mann, W.C. Inquiry Semantics: A Functional Semantics of Natural Language Grammar. USC/ISI Research Report RR-83-8.

[Mann 83] Mann, W.C. A Linguistic Overview of the Nigel Text Generation Grammar. USC/ISI Research Report RR-83-9.

[Mann & Matthiessen 83] Mann, W.C. and Matthiessen, C.M.I.M. Nigel: A Systemic Grammar for Text Generation. In *Systemic Perspectives on Discourse: Selected Papers Papers from the Ninth International Systemics Workshop*, Benson, R. and Greaves, J. (eds), Ablex: London, 1985. Also available as USC/ISI Research Report RR-83-105.

[Matthiessen 84] Matthiessen, C.M.I.M. Systemic Grammar in Computation: The Nigel Case. In *Proceedings of 1st Conference of the European Association for Computational Linguistics*, Pisa, 1983. Also available as USC/ISI Research Report RR-84-121, 1984.

[Matthiessen 87a] Matthiessen, C.M.I.M. Notes on the Organization of the Environment of a Text Generation Grammar. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen, G. (ed), Nijhoff, 1987. Also available as USC/ISI Research Report RR-87-177, 1987.

[Matthiessen 87b] Matthiessen, C.M.I.M. Semantics for Systemic Grammar: The Chooser and Inquiry Framework. In *Systemic Perspectives on Discourse*, 1987. Also available as USC/ISI Research Report RR-87-189, 1987.

[Nakamura et al. 88] Nakamura, J., Tsujii, J. and Nagao, M. GRADE: A Software Environment for Machine Translation. In *Computers and Translation*, Vol. 3:1, pp. 69-82, 1988.

[Nirenburg 87] Nirenburg, S. (ed). *Machine Translation: Theoretical and Methodological Issues.* Cambridge University Press, Cambridge, 1987.

[Patten 88] Patten, T. *Systemic Text Generation as Problem Solving.* Cambridge University Press, Cambridge, 1988.

[Penman 88] *The Penman Primer, User Guide, and Reference Manual.* Unpublished USC/ISI documentation, 1988.

[Pollard & Sag 87] Pollard, C. and Sag, I. *Information Based Syntax.* CSLI Lecture Notes Number 13, University of Chicago Press, 1987.

[Rounds & Kasper 86] Rounds, W. and Kasper, R.T. A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, Cambridge, MA, June 1986.

[Shieber 84] Shieber, S.M. The design of a computer language for linguistic information. In *Proceedings of the Tenth International Conference on Computational Linguistics: COLING 84*, Stanford University, Stanford, 1984.

[Sondheimer et al. 84] Sondheimer, N.K., Weischedel, R.M. and Bobrow, R.J. Semantic Interpretation Using KL-ONE. In *Proceedings of the Tenth International Conference on Computational Linguistics: COLING 84*, Stanford, CA, July 1984.

[Winograd 72] Winograd, T. *Understanding Natural Language.* Academic Press, New York, 1972.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SERCURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILTY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(s)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: USC/Information Sciences Institute; 6b. OFFICE SYMBOL	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): 4676 Admiralty Way, Marina del Rey, CA 90292	7b. ADDRESS (City, State, and Zip Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research; 8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: MDA-903-87-C-0641
8c. ADDRESS (City, State, and ZIP Code): Universtiy of Southern Calif at San Diego (A-034), Scripps Institute of Oceanography, 8603 LaJolla Shores Drive, San Diego, CA 92093-0234	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (INCLUDE SECURITY CLASSIFICATION)

Natural Language

12. PERSONAL AUTHOR(S)

Eduard H. Hovy

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF EPORT (Year, Month,Day)	15. PAGE COUNT
Final	FROM ____ TO ____		

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identfy by block number)
FIELD	GROUP	SUB-GROUP	natural language processing, computational linguistics, language generation, text planning, parsing, machine translation, computational grammars, Systemic Functional Linguistics, Rhetorical Structure Theory, Penman

19. ABSTRACT (Continue on reverse if necessary and identfy by block number)

The report describes research and development of natural language generation and some parsing theory and technology in the Penman project of the Information Sciences Institute of Southern California. Penman, set of a natural language generation systems, provides computational technology for generating English sentences and paragraphs, starting with input specifications of a non-linguistic kind. Its associated information resrouces, such as high-level concept models, domain models, lexicons, etc., are also employed for other uses, including information retrieval and machine translation.

The following five areas of research and development were addressed during the funding period being reported on in this document:

- Sentence generation: continued expansion and distribution of the Penman sentence generation system
- Multisentence text planning: development and testing of new text structure planning technology
- Parsing: construction of a prototype parser using the Penman system's grammar and semantic models
- Information retrieval: conceptual outline of an IR system that leverages off Penman's semantic knowledges recources
- Machine translation: development work toward the eventual coolaboration with two other sites in the joint construction of a new MT system

continued on back

20. DISTRIBUTION/AVAILBILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL		22b TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL